

AMENDMENTS TO THE CLAIMS

Please amend the Claims as follows:

Please cancel claims 1-26.

Please add new claims 27-41.

27. (Newly Added) A computer-based method for compiling a structured document schema into type annotation records comprising steps of:

a. building a type hierarchy ordered tree from structured document with each type record node in said hierarchy ordered tree containing a typing tuple of the following format:

<string_value, type>;

b. creating a complete typing SET which contains all typing tuples in said type hierarchy ordered tree;

c. sorting said typing tuples in said SET by their first field, *string_value*;

d. creating, from sorted tuples in (c), ambiguity typing sequences for tuples having a common first field, *string_value*;

e. sorting and patching ambiguity typing sequences with dummy items;

f. creating a typing array by concatenating typing tuples in said sorted and patched ambiguity typing sequences;

g. for each type record node, N, in created typing array, if the intersection of a set of tuples in N with any ambiguity typing sequence is not empty, then replacing first typing tuple *<string_n, type_n>* in N by *(string_n, type_n, offset>*, wherein offset represents a position of an ambiguity type in a given ambiguity typing sequence;

h. creating an index structure to link each *string_value* to its type; and

- i. outputting said created index structure.

28. (Newly Added) The computer-based method of claim 27, wherein said structured document schema is an XML document schema.

29. (Newly Added) The computer-based method of claim 27, wherein said *string_value* is any of the following: a type name, element name or attribute name.

30. (Newly Added) The computer-based method of claim 27, wherein said index structure is any of the following: hash table, binary tree, or B+ tree.

31. (Newly Added) The computer-based method of claim 27, wherein said computer-based method is implemented in a database.

32. (Newly Added) The computer-based method of claim 27, wherein said sorting of said typing tuples in said SET by their first field, *string_value* is based an alphabetical sort.

33. (Newly Added) An article of manufacture comprising a computer usable medium having computer readable program code embodied therein which implements a method for compiling a structured document schema into type annotation records, said computer usable medium comprising steps of:

- a. computer readable program code building a type hierarchy ordered tree from structured document with each type record node in said hierarchy ordered tree containing a typing tuple of the following format: *<string_value, type>*;
- b. computer readable program code creating a complete typing SET which contains all typing tuples in said type hierarchy ordered tree;
- c. computer readable program code sorting said typing tuples in said SET by their first field, *string_value*;
- d. computer readable program code creating, from sorted tuples in (c), ambiguity typing sequences for tuples having a common first field, *string_value*;
- e. computer readable program code sorting and patching ambiguity typing sequences with dummy items;
- f. computer readable program code creating a typing array by concatenating typing tuples in said sorted and patched ambiguity typing sequences;
- g. computer readable program code for each type record node, N, in created typing array, if the intersection of a set of tuples in N with any ambiguity typing sequence is not empty, then replacing first typing tuple *<string_n, type_n>* in N by *(string_n, type_n, offset)*, wherein offset represents a position of an ambiguity type in a given ambiguity typing sequence;
- h. computer readable program code creating an index structure to link each *string_value* to its type; and
- i. computer readable program code outputting said created index structure.

34. (Newly Added) The article of manufacture of claim 33, wherein said structured document schema is an XML document schema.

35. (Newly Added) The article of manufacture of claim 33, wherein said *string_value* is any of the following: a type name, element name or attribute name.

36. (Newly Added) The article of manufacture of claim 33, wherein said index structure is any of the following: hash table, binary tree, or B+ tree.

37. (Newly Added) The article of manufacture of claim 33, wherein said arranging step is further comprised of: collecting each third field of said typing tuples and sorting said typing tuples in said ambiguity sequence with respect to third field of said typing tuple.

38. (Newly Added) The article of manufacture of claim 33, wherein said sorting of said typing tuples in said SET by their first field, *string_value* is based an alphabetical sort.

39. (Newly Added) A computer-based method for compiling a structured document schema into type annotation records comprising steps of:

a. building a type hierarchy ordered tree from XML document schema with each type record node in said hierarchy ordered tree containing a typing tuple of the following format:
<*string_value*, *type*>;

b. creating a complete typing SET which contains all typing tuples in said type hierarchy ordered tree;

c. alphabetical sorting said typing tuples in said SET by their first field, *string_value*;

d. creating, from sorted tuples in (c), ambiguity typing sequences for tuples having a common first field, *string_value*;

- e. sorting and patching ambiguity typing sequences with dummy items;
- f. creating a typing array by concatenating typing tuples in said sorted and patched ambiguity typing sequences;
- g. for each type record node, N, in created typing array, if the intersection of a set of tuples in N with any ambiguity typing sequence is not empty, then replacing first typing tuple $\langle string_n, type_n \rangle$ in N by $\langle string_n, type_n, offset \rangle$, wherein offset represents a position of an ambiguity type in a given ambiguity typing sequence;
- h. creating any of the following index data structures to link each string_value to its type: hash table, binary tree, or B+ tree; and
- i. outputting said created index structure.

40. (Newly Added) The computer-based method of claim 39, wherein said *string_value* is any of the following: a type name, element name or attribute name.

41. (Newly Added) The computer-based method of claim 39, wherein said computer-based method is implemented in a database.